

AD-A270 976

PSL-93/49



2

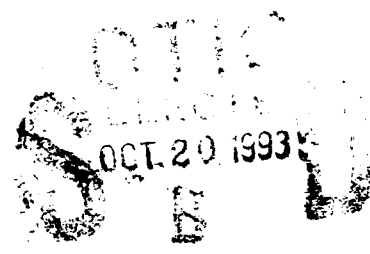
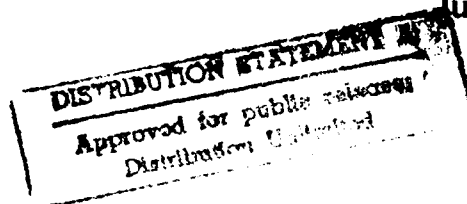
Systolic Design with Asynchronous Controls for Digital-Signal Processing

Final Report

Author:

Dr. Kamran Reihani

July 15, 1993



U.S. Army Research Office
Grant: DAAL03-90-G-0211



Physical Science Laboratory

P.O. Box 30002

Las Cruces, NM 88003-0002

(505) 522-9100 FAX 505-522-9389/9434

93-25093



93 10 10 116

**Systolic Design with Asynchronous Controls for
Digital-Signal Processing**

Final Report

Author:

Dr. Kamran Reihani

July 15, 1993

**U. S. Army Research Office
Grant: DAAL03-90-G-0211**

**Physical Science Laboratory
New Mexico State University
Box 30002
Las Cruces, New Mexico 88003-0002**

The views, opinions, and/or findings contained in this report are those of the author and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

REPORT DOCUMENTATION PAGE

Form Approved

OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204 Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE 07/15/93		3. REPORT TYPE AND DATES COVERED Final: 08/08/91 - 07/15/93	
4. TITLE AND SUBTITLE Systolic Design with Asynchronous Controls for Digital-Signal Processing				5. FUNDING NUMBERS DAAL03-90-G-0211	
6. AUTHOR(S) Dr. Kamran Reihani					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Physical Science Laboratory New Mexico State University Box 30002 Las Cruces, NM 88003-0002				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Office P. O. Box 12211 Research Triangle Park, NC 27709-2211				10. SPONSORING / MONITORING AGENCY REPORT NUMBER ARO 2876.9-MA SAT	
11. SUPPLEMENTARY NOTES The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.					
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited.				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The research sponsored by this grant is focused on the development of a theoretical and technological basis for designing efficient systolic arrays for digital-signal processing algorithms. The major contributions of the research are the following: <ul style="list-style-type: none"> - Data reduction techniques via the utilization of algorithm properties - Conversion of sequential input signal into input blocks by means of a spiral systolic mesh that is suitable for parallel processing and that is flexible for enabling various array dimensions - Devising a hybrid of SA and data-flow approach, making data streams independent of computations executed in each processor, thus reducing waiting time. The communication (PE) protocols resolve the data-flow conflicts created by the merging of the spiral and asynchronous systolic array architecture 					
14. SUBJECT TERMS Asynchronous controls, Block kernel cycle, Communication protocol, Computational graph, Connection matrices, Data propagation, Digital-Signal processing algorithms, Processing element, Spiral mesh, Systolic array				15. NUMBER OF PAGES 31 pages	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED		18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED		19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	
				20. LIMITATION OF ABSTRACT UL	

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to **stay within the lines** to meet optical scanning requirements.

Block 1. Agency Use Only (Leave blank).

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C - Contract	PR - Project
G - Grant	TA - Task
PE - Program Element	WU - Work Unit Accession No.

Block 6. Author(s) Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es) Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. (If known)

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."

DOE - See authorities.

NASA - See Handbook NHB 2200.2.

NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.

DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.

NASA - Leave blank.

NTIS - Leave blank.

Block 13. Abstract. Include a brief (Maximum 200 words) factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code (NTIS only).

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

TABLE OF CONTENTS

	Page No.
LIST OF FIGURES	2
1. STATEMENT OF RESEARCH PROBLEM	3
2. SUMMARY OF RESULTS	4
2.1 Theoretical Aspect	4
2.2 System Aspect	6
2.2.1 Spiral Architecture	8
2.2.2 Asynchronous Architecture	13
2.3 Performance Aspect	18
2.4 References	19
3. PUBLICATIONS AND REPORTS	21
4. PERSONNEL SUPPORTED AND DEGREES AWARDED	22
APPENDIX A	23
APPENDIX B	25
APPENDIX C	27
APPENDIX D	29

DTIC QUALITY INSPECTION

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
List	Special
A-1	

LIST OF FIGURES

	Page No.
Fig. 1. One-step adaptive lattice predictor, $L = 3$	5
Fig. 2. Computational graph for $\varepsilon_{f,k}$ and $\varepsilon_{b,k}$	7
Fig. 3. Computational scheme for $\varepsilon_{f,k}$ and $\varepsilon_{b,k}$	7
Fig. 4. Interconnection scheme for $\varepsilon_{f,k}$ and $\varepsilon_{b,k}$	7
Fig. 5. Proposed PE_k for $\varepsilon_{f,k}$ and $\varepsilon_{b,k}$	7
Fig. 6. Propagation of data between array PEs	8
Fig. 7. Spiral systolic array architecture for LPF filter of order 12	9
Fig. 8. Inputs and outputs of a PE	10
Fig. 9. PE input/output data distribution for kernel cycle $t = 9$	12
Fig. 10. Proposed protocol in (type I) $PE_{i,j}$	17

1. STATEMENT OF RESEARCH PROBLEM

The systolic algorithm approach is currently the most effective design procedure for minimizing computing time and the number of processors. Systolic arrays (SAs) are examples of VLSI special purpose processor networks that realize computationally expensive algorithms in hardware, and can achieve massive concurrency. Such arrays are numerous used for the real-time implementation of digital-signal processing algorithms, because of the properties that these algorithms possess for their effective mapping onto high-speed SA architectures. The main problem in a systolic approach is that extra delays are needed to assure proper timing and synchronization, which will collectively slow down computation and, thereby, reduce the throughput rate. For a large-scale array, this synchronization can become particularly tedious. Moreover, algorithms for asynchronous SAs (i) have not been fully utilized for data reduction, and (ii) the conversion of sequential input signals into input blocks that will alter the organization of the systolic mesh has not been performed.

The research sponsored by this grant focused on the following important technical issues that result in the development of efficient SAs.

- Data reduction techniques via the utilization of algorithm properties.
- Conversion of sequential input signals into input blocks by means of a spiral systolic mesh that is suitable for parallel processing and that is flexible for enabling various array dimensions.
- Making data streams independent of computations executed in each processor, thus reducing waiting time.

The main objective of this research grant is to develop a theoretical and technological basis for designing an asynchronous SA — that is a hybrid of SA and data-flow approach — one capable of converting input signals into input blocks for producing a faster and more flexible system.¹ The key to this design is a spiral SA structure, self-timed processors, and communication protocols to get control of data streams so that each computation can start if all its data are available. Moreover, the communication protocols resolve the data-flow conflicts created by the merging of the spiral and asynchronous SA architectures. The SA processes the input signal efficiently and eliminates the complex shift register organization of traditional filter realizations. Incorporated in this design are maximum parallelism and pipelinability, trade-off among computations, communications, and memory. Furthermore, the systolic array will use simple local interconnections without undesirable properties, such as preloading input data or global broadcasting.

¹ For most digital signal processing algorithms.

2. SUMMARY OF RESULTS

The research performed during this grant focused on the improvement of present methods of designing systolic arrays [1-7] for the following digital signal processing algorithms [8-15]:

- discrete Fourier transform (DFT)
- inverse discrete Fourier transform (IDFT)
- convolution
- correlation
- linear phase filter (LPF)
- discrete Hadamard transform (DHT)
- one-step adaptive lattice predictors
- arbitrarily large LMS adaptive filters

This required a comprehensive research effort that addresses theoretical, system, and performance aspects.

Theoretical Aspect: Describe useful mathematical representations for digital-signal processing (DSP) algorithms.

System Aspect: Develop a technique for tailoring the algorithms into forms suitable for mapping onto SA architectures and to develop array systems for implementing the architectures efficiently.

Performance Aspect: Evaluate the performance of the new algorithms developed especially for the asynchronous array.

2.1 Theoretical Aspect

The mathematical representations for DSP algorithms include important properties such as local and regular data broadcasting. On the basis of these representations, we have arrived at a design procedure for creating a systolic array and a systolic algorithm for the realization of DSP algorithms efficiently. For example, lattice filters offer the following features:

1. They are very efficient structures, because of their simultaneous production of the forward and backward prediction errors.
2. The various stages of the lattice predictors are decoupled from each other. Moreover, the backward prediction errors generated by the stages of a lattice predictor are orthogonal to each other for wide-sense stationary input data.
3. They are modular in structure; an order increase is induced by adding one or more stages without affecting previous computations.
4. The structure of each of the stages is identical, making lattice filters viable candidates for VLSI implementation.

Figure 1 illustrates the diagram of a one-step predictor of order 3.

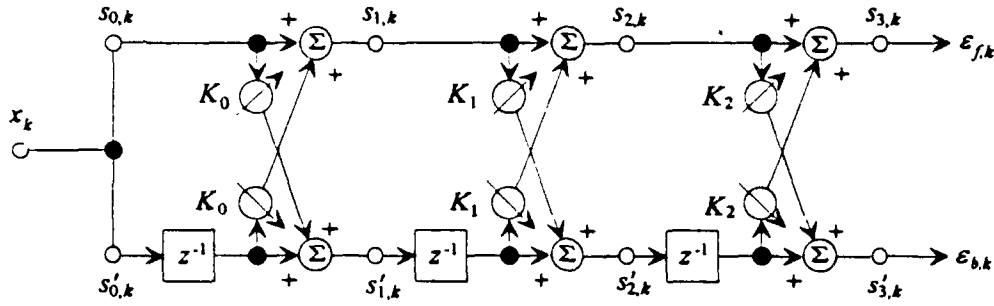


Fig. 1. One-step adaptive lattice predictor, $L = 3$.

The forward and backward signals are labeled $s_{l,k}$ and $s'_{l,k}$, respectively. The relations describing a **one-step adaptive lattice predictor** are

$$\begin{aligned} s_{0,k} &= s'_{0,k} = x_k \\ s_{l+1,k} &= s_{l,k} + K_{l,k} s'_{l,k-1} \quad 0 \leq l \leq L-1 \\ s'_{l+1,k} &= K_{l,k} s_{l,k} + s'_{l,k-1} \quad 0 \leq l \leq L-1 \\ \varepsilon_{f,k} &= s_{L,k} \\ \varepsilon_{b,k} &= s'_{L,k} \end{aligned} \quad (1)$$

where, $\varepsilon_{f,k}$ and $\varepsilon_{b,k}$ are the respective forward and backward predicted errors. The LMS algorithm for this lattice is

$$K_{l,k+1} = K_{l,k} - 2\mu_l s_{l+1,k} s'_{l,k-1} \quad 0 \leq l \leq L-1 \quad (2)$$

Here, μ_l is the so-called convergence parameter, which is time invariant. This parameter is set to different values from stage to stage [13]. The **data reduction technique** can be demonstrated via the theoretical aspect of the **LPF digital filters**. The one-dimensional FIR filter is mathematically described as [11]

$$y(k) = \sum_{n=1}^N \omega(n) x(k-n+1) \quad k = 1, 2, \dots, N+L-1 \quad (3)$$

where $y(k)$ is the filter output, $x(m)$ is the filter's input signal, $m = 1, 2, \dots, L$, and $\omega(n)$ is the filter's impulse response.

For LPF digital filters, the following four cases are considered.

Case 1, 2: Even filter order (+ for even and - for odd symmetry):

$$\begin{aligned} y(k) &= \sum_{n=1}^{\frac{N}{2}} \omega(n) [x(k-n+1) \pm x(k-N+n)] \\ \omega(n) &= \pm \omega(N+1-n) \end{aligned} \quad (4)$$

Case 3, 4: Odd filter order (+ for even and - for odd symmetry):

$$\begin{aligned}
 y(k) &= \sum_{n=1}^{\frac{N-1}{2}} \omega(n) [x(k-n+1) \pm x(k-N+n)] + u(k) \\
 \omega(n) &= \pm \omega(N+1-n) \\
 u(k) &= \begin{cases} \omega\left(\frac{N+1}{2}\right)x\left(k-\frac{N-1}{2}\right) & \text{for even symmetry} \\ 0 & \text{for odd symmetry} \end{cases}
 \end{aligned} \tag{5}$$

Because of the similarities between the first two and last two cases, case 1 and case 2 are combined into one class, and case 3 and case 4 are combined into another class, with the respective reduced filter orders of $N/2$, $N/2$, $(N+1)/2$, and $(N-1)/2$ for SA implementation [16].

A summary of the theoretical aspect for the remaining DSP algorithms is listed in Appendix A.²

2.2 System Aspect

A fundamental problem in the design of SA structures is to obtain localized algorithms, namely, localization of data dependencies. To achieve maximum parallelism in an algorithm, data dependencies during the computations must be determined. The following notations can be useful for the system aspect of SA design:

Definition 1: A dependency graph shows the dependency of the computations that occur in an algorithm. An algorithm is computable if, and only if, its dependency graph contains no loops or cycles.

Definition 2: A localized dependency graph has only local dependencies, i.e., the length of each dependency arc is independent of the problem size.

Definition 3: A computational graph is a localized dependency graph with each node in the graph labeled by the indices of the terms it computes.

For SA architecture, determining the computational graph for a given algorithm is a suitable starting point for non-structurally altered arrays. For example, spiral SAs have been structurally altered to conform to an architecture capable of processing blocks of input signals, rather than sequential input signals. The computational graph for the one-step adaptive lattice predictor is shown in Fig. 2. The large circles represent processing elements (PEs), and the first and second digits of the small circles represent one of nine operations, shown in Fig. 3, and the PE number, respectively.

² This list is presented to maintain continuity with the system and performance design aspects of the DSP algorithms.

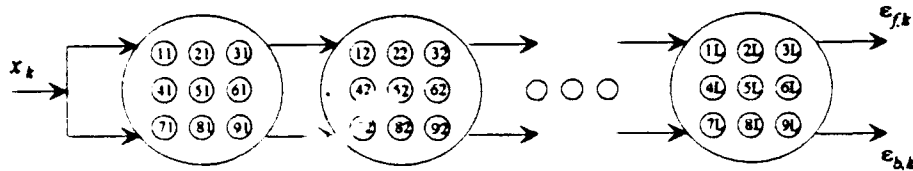


Fig. 2. Computational graph for $e_{f,k}$ and $e_{b,k}$.

The computational scheme illustrates identical sets of operations in various stages. This enables us to determine the interconnection scheme which reveals the PE organization and PEs' input and output. The computational and interconnection scheme for the one-step adaptive lattice predictor are illustrated in Figs. 3 and 4, and the internal operations of a PE are shown in Fig. 5 [17]. The system aspect of the SA design for the **DSP algorithms** is listed in Appendix B [18-19].

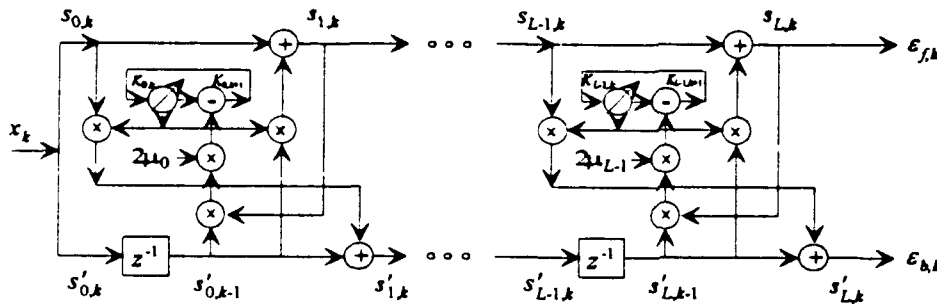


Fig. 3. Computational scheme for $e_{f,k}$ and $e_{b,k}$.

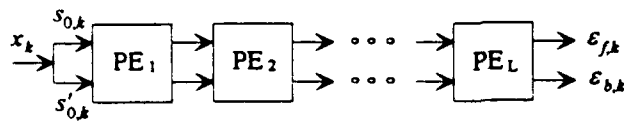


Fig. 4. Interconnection scheme for $e_{f,k}$ and $e_{b,k}$.

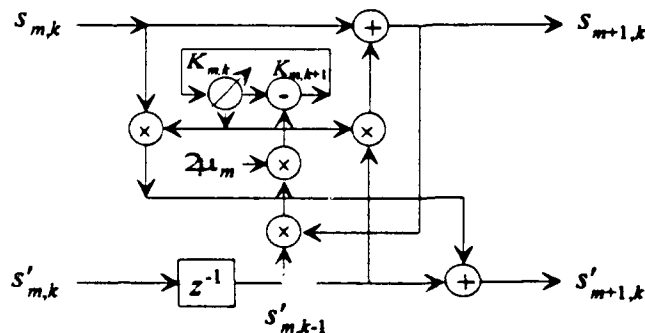


Fig. 5. Proposed PE_k for $e_{f,k}$ and $e_{b,k}$.

2.2.1 Spiral Architecture

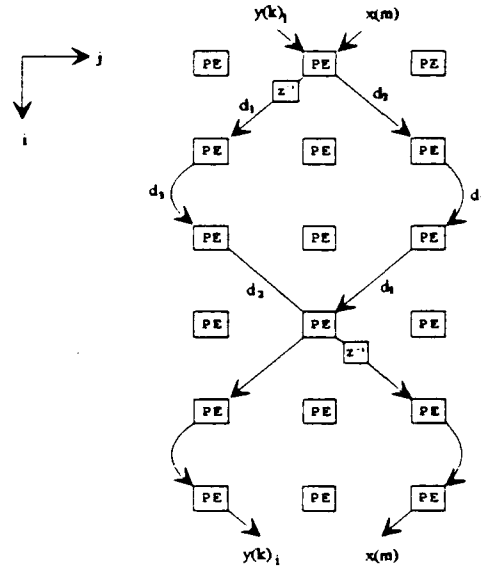
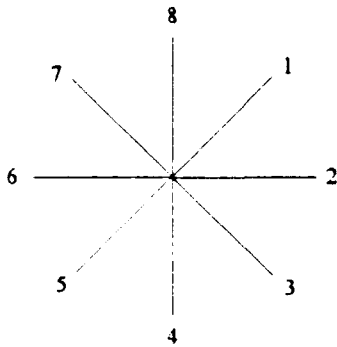
To illustrate this method, we describe herein the process involved in designing a spiral SA for an LPF digital filter: An input signal $x(m)$ is passed from the input x_{in} of a PE at (i, j) to its output x_{out} (see equations 4 and 5). Using zero or one delay, this input is transferred to the input x_{in} of the neighboring cell $(i + 1, j^*)$. The index j^* can be identified as $j + 1$ or $j - 1$ using an analogy from optics. Input data $x(k)$ moves between the PEs in the direction $d_1 = 5$ or $d_2 = 3$ until it reaches the left or right boundary of the array, respectively (see Fig. 6(a) and (b)). Then the input is "reflected" back in the respective direction $d_1 = 3$ or $d_2 = 5$, with the intermediate direction $d_3 = 4$. The intercell propagation of intermediate values $y(k)$, of input data $y(k)$ can be described by

$$y(k)_i = y(k)_{i-1} + w(i) \cdot x(k - i + 1) \quad (6)$$

where, $y(k)_0 = 0$ and $y(k)_i = y(k)$. A delay element is positioned immediately following a PE at (i, j) , provided that

$$j = \begin{cases} (i + 1)_{\text{mod } M} \\ (i)_{\text{mod } M} \\ (M - i)_{\text{mod } M} \\ (M - i + 1)_{\text{mod } M} \end{cases} \quad \text{for } \begin{cases} (r - 1)M \leq i < rM & r \text{ odd} \\ i = rM \\ (r - 1)M \leq i < rM & r \text{ even} \\ i = rM \end{cases} \quad (7)$$

where, M is the number of input data blocks.



(a) Directional encoded values

(b) Data propagation between array PEs

Fig. 6. Propagation of data between array PEs.

The two systolic array architectures with spiral structure of intercell connections, proposed to compute equation (4) for class 1 and equation (5) for class 2 (for $N = 12$) are depicted in Fig. 7(a) and (b), respectively [16]. The secondary input

signal $x(k - N + n)$ is shown integrated into the structure. In this structure, the PE performs the following operation, as illustrated in Fig. 8:

$$\begin{aligned} x_{out} &= x_{in} \\ y_{out} &= y_{in} + \omega(i) [x_{in} \pm x'_{in}] \end{aligned} \quad (8)$$

Here $x(k - N + n)$ is the past input data, symmetric to $x(k - n + 1)$, and $N - 2i + 1$ sampling periods apart.

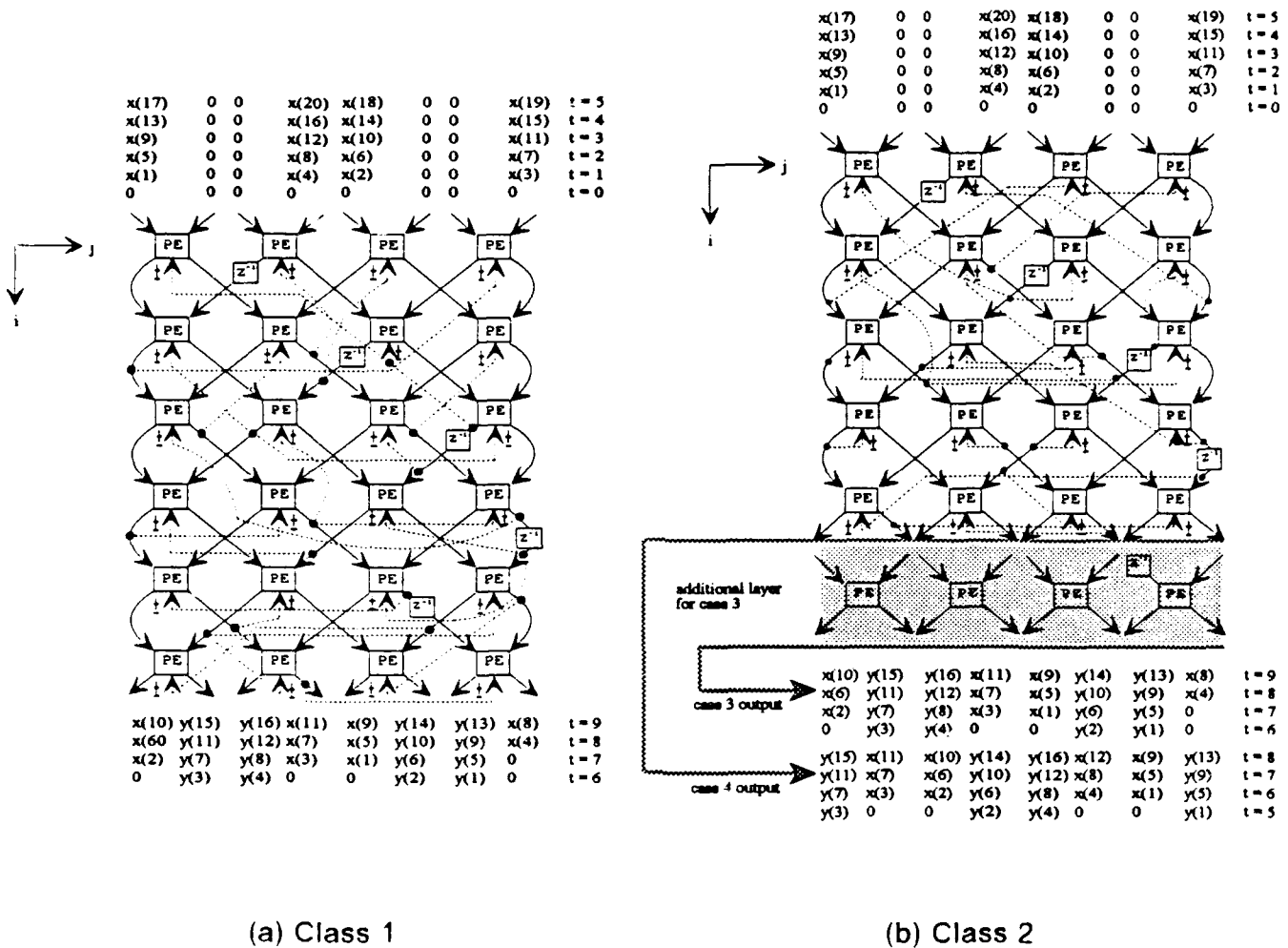


Fig. 7. Spiral systolic array architecture for LPF filter of order 12.
(Notations: + for even symmetry and - for odd symmetry.)

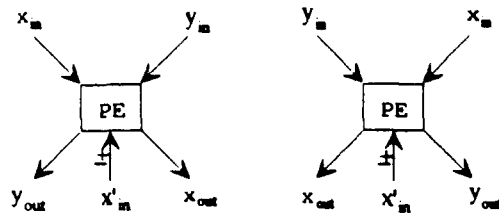


Fig. 8. Inputs and outputs of a PE.

Let U , the reduced filter order, and M , the number of input data blocks, designate the spiral SA's number of rows and columns, respectively. Denote $\text{indx}_i(i, j, t)$ and $\text{indx}_o(i, j, t)$ to be the respective indices of input and output signals of a PE, where i, j , and t are the row, column and block kernel cycle. Here $i = 1, \dots, U$, $j = 1, \dots, M$, and $t = 1, \dots, T_{\max}$ ($T_{\max} = (K/M + .5)_{\text{integer}}$), where K is the number of input signals. Note that the indices of input and output signals are related by the following:

$$\text{indx}_o(i, j, t) = \text{indx}_i(i, j, t) - M \quad (9)$$

The following algorithm demonstrates the positioning of the input signals in the PEs of the spiral SA, at kernel cycle t .

Algorithm 1. { * **Input:** spiral SA's row and column size, and input data size
Output: input signals' indices, $\text{indx}_i(i, j, t)$. * }

Begin

$T_{\max} = (K/M + 0.5)_{\text{integer}}$

for $t = 1$ to T_{\max} do

begin

$R = \begin{cases} M/2, & \text{if } M \text{ even} \\ (M-1)/2, & \text{if } M \text{ odd} \end{cases}$

$i = 1;$

for $j = 1$ to M do { 1st row computations }

begin

$\text{indx}_i[i, j, t] = \begin{cases} (t-1)M + (j+1)/2, & \text{if } j \text{ odd} \\ tM - (j-2)/2, & \text{if } j \text{ even} \end{cases}$

end;

for $i = 2$ to N do { rows 2 to N computations }

begin

$p = i \bmod 2;$

for $j = 1$ to M do

$\text{indx}_i[i, j, t] = \text{indx}_i[i-1, j, t] - M - 1;$

for $q = 1$ to $\begin{cases} R-1, & \text{if } i \text{ even} \\ R, & \text{if } i \text{ odd} \end{cases}$ do

begin

$j1 = 2q + p - 2; \quad j2 = j1 + 1;$

if $j2 \leq M$ then

begin

$\text{mid} = \text{indx}_i[i, j1, t];$ { swap of indices }

$\text{indx}_i[i, j1, t] = \text{indx}_i[i, j2, t];$

$\text{indx}_i[i, j2, t] = \text{mid};$

end; { $j2 \leq M$ }

end;

end; { $i = 2, N$ }

end; { $t = 1, T_{\max}$ }

End.

The index of some feedback input, ℓ , is related to the index of its destination PE, k by (see eq. 4 and Fig. 7).

$$\begin{aligned}\ell &= k + \Delta k\ell \\ \Delta k\ell &= (k - N + n) - (k - n + 1) = -N + 2n - 1 \\ \ell &= k + 2n - N - 1\end{aligned}\tag{10}$$

An example for the PE input/output data distribution in the spiral SA, with $N/2 = 6$ and $M = 4$ dimensions (at kernel cycle $t = 9$) is shown in Fig. 9:

PE input output data distr.	$j = 1$	$j = 2$	$j = 3$	$j = 4$
$i = 1$	$\frac{x(33)}{x(29)}$	$\frac{x(36)}{x(32)}$	$\frac{x(34)}{x(30)}$	$\frac{x(35)}{x(31)}$
$i = 2$	$\frac{x(28)}{x(24)}$	$\frac{x(29)}{x(25)}$	$\frac{x(31)}{x(27)}$	$\frac{x(30)}{x(26)}$
$i = 3$	$\frac{x(24)}{x(20)}$	$\frac{x(23)}{x(19)}$	$\frac{x(25)}{x(21)}$	$\frac{x(26)}{x(22)}$
$i = 4$	$\frac{x(19)}{x(15)}$	$\frac{x(20)}{x(16)}$	$\frac{x(18)}{x(14)}$	$\frac{x(21)}{x(17)}$
$i = 5$	$\frac{x(15)}{x(11)}$	$\frac{x(14)}{x(10)}$	$\frac{x(16)}{x(12)}$	$\frac{x(13)}{x(9)}$
$i = 6$	$\frac{x(10)}{x(6)}$	$\frac{x(11)}{x(7)}$	$\frac{x(9)}{x(5)}$	$\frac{x(8)}{x(4)}$

Fig. 9. PE input/output data distribution for kernel cycle $t = 9$.

The connection matrices for the feedback input data of the class 1 and class 2 spiral SA (LPF of order 12) are shown here. (Note that $\frac{N}{2} = 6$, $M = 4$ for class 1 and $\frac{(N-1)}{2} = 5$, $M = 4$ for class 2.)

$$\text{class 1: } \begin{bmatrix} \pm 0(3, 4) & \pm 0(2, 2) & \pm i(3, 2) & \pm 0(2, 1) \\ \pm 0(3, 2) & \pm 0(3, 1) & \pm 0(3, 4) & \pm 0(3, 3) \\ \pm 0(4, 4) & \pm 0(4, 2) & \pm i(4, 3) & \pm 0(3, 2) \\ \pm 0(4, 3) & \pm 0(4, 1) & \pm i(5, 4) & \pm 0(4, 2) \\ \pm 0(5, 3) & \pm 0(5, 1) & \pm i(5, 4) & \pm 0(5, 2) \\ \pm 0(5, 4) & \pm 0(5, 2) & \pm i(6, 4) & \pm 0(6, 2) \end{bmatrix}\tag{11}$$

$$\text{class 2: } \begin{bmatrix} \pm i(3, 2) & \pm 0(2, 4) & \pm 0(2, 1) & \pm 0(2, 2) \\ \pm 0(3, 1) & \pm 0(3, 3) & \pm i(3, 2) & \pm 0(3, 4) \\ \pm i(4, 3) & \pm 0(4, 4) & \pm 0(3, 2) & \pm 0(3, 1) \\ \pm 0(4, 1) & \pm 0(4, 2) & \pm 0(4, 3) & \pm 0(4, 4) \\ \pm i(5, 4) & \pm 0(5, 3) & \pm i(5, 2) & \pm 0(5, 1) \end{bmatrix} \quad (12)$$

where $0(i, j)$ and $i(i, j)$ at the (m, n) location of the connection matrix refer to the connection made from the respective output and input of the PE at location (i, j) to the input of the PE that makes up the processor's feedback input data at the location (m, n) of the spiral SA.

The spiral SA structure for computing eqs. A4 and A5 for the DFT and IDFT algorithms and for computing eqs. A2 and A3 for the convolution and correlation algorithms are illustrated in Appendix C [20]. Note that contrary to the LPF spiral SA structure neither structure requires a feedback input, and in addition, the DFT and IDFT spiral SA structures consist of no delay elements.

2.2.2 Asynchronous Architecture

The performance of a systolic array can be improved further if an asynchronous approach is adopted. The idea is to design self-timed processors and communication protocols to gain control of data streams, so that each computation can start if all its data are available. The proposed array is a hybrid of a systolic array and a data-flow machine.

In the asynchronous design, instead of using a global clock, self-timing PEs and a communication protocol are employed. The advantage is the following: The whole period of a clock unit for addition/subtraction, multiplication, addition, and data routing can be separated into several small steps, and some of these steps can be executed simultaneously. The concept of asynchronous computations can be specified as shown below, with steps 2 and 3 executed simultaneously.

Step 1

- send/receive the respective acknowledge signal and data to the previous processor
- send a request signal that simultaneously forwards data to the next processor

Step 2

- transfer data to the next processor

Step 3

- add/subtract, multiply, and accumulate the results

The basic features of the proposed array remain the same as in the systolic array, with the exception that the data routing and computing in each PE can be operated simultaneously. Moreover, a PE does not wait for data until the previous PE completes its computations. The following algorithm reflects this new feature for an LPF digital filter [16].

Algorithm 2: Linear Phase FIR Digital Filter (Asynchronous)

Begin

While there are data entering PE, **do**

Begin

Receive input data x_{in} , x'_{in} , y_{in} ;

Send x_i to the next processing element ;

& $v_{i,j} \leftarrow x_{in} \pm x'_{in}$;

$u_{i,j} \leftarrow \omega_i v_{i,j}$;

$y_{out} \leftarrow u_{i,j} + y_{in}$;

End While

End

Protocol realization for Algorithm 2

A protocol for realization of Algorithm 2 must control the flow of data and make the data flow independent of the internal operations in each PE, such that the values of input variables are not overwritten during their computing time intervals. *Because in spiral SAs timing is crucial for the synchronization of the PEs' input data, and in asynchronous machines operations are triggered by the availability of the data, the system protocol for Algorithm 2 (1) triggers PEs' operations by the fullness of their x_{in} , x'_{in} , and y_{in} input ports, and (2) allows data routing and computing to be performed simultaneously.* In the proposed protocol for an LPF digital filter, five types of signals — REQ, ACK, FLG, EMP, FUL — are introduced, two of which are external signals and three of which are internal signals. The REQ signal reports to the next PE that the data in its output port are ready for transmission. The ACK signal reports to the previous PE that its input port is ready to receive new data. The FLG, EMP, and FUL signals indicate the completeness of the add/subtract-multiply-and-add computations, and the emptiness and fullness of the input port(s), respectively. The protocol can be described formally for data x_{in} as shown below, while noting that the specifications of the protocols for x'_{in} and y_{in} are similar.

1. PE_{i,j} receives a request, REQ ^{x_{in}} from a source PE when the data in its output port are ready to be transmitted.

2. $PE_{i,j}$ receives an acknowledge ACK^{xin} from a destined PE when its input port is ready to accept new data.
3. $PE_{i,j}$ has three internal signals, FLG^{xin} , EMP^{xin} , and FUL^{xin} , which indicate the completeness of the add/subtract-multiply-and-add operations, and the emptiness and fullness of the input port(s).
4. $PE_{i,j}$ contains the following transition latches:

Latch 1 activates the signal ACK^{xin} , which is fired toward a source PE only after both signals REQ^{xin} and EMP^{xin} are received.

Latch 2 controls the data flow from the input port to the buffer-for-add/subtract and to the output port, and it also activates the signals EMP^{xin} and FUL^{xin} . It is fired only after the signal FLG^{xin} is received.

Latch 3 controls the data flow from the output port of the PE to the input port of a destined PE. It is fired only after the signal ACK^{xin} is received from the destined PE.

A detailed configuration of this protocol for type I cell (case 1 or 3 FIR linear phase digital filters) is depicted in Figure 10. In this configuration, there are two lines that intersect each PE. One is a bidirectional control line for the transmission of the ACK and REQ bit-signals, and the other is a unidirectional data line. The linear phase FIR digital filter operations in $PE_{i,j}$ are shown in Algorithm 3 [16].

The PE protocols for some of the most important **DSP algorithms** are shown in Appendix D [17-19].

**Algorithm 3: Linear Phase FIR Digital Filter Operation
(A Completed Asynchronous Version)**

```

Begin
  While there are data entering  $PE_{i,j}$  do
    Begin
      For all  $PE_{i,j}$ s asynchronously do
        Begin
          Wait for  $REQ_{input}$  from a source PE ;
          Receive  $REQ_{input}$  ;
          While input port is empty do
            Begin
              Activate  $REQ_{input}$  & EMP ;
              Send  $ACK_{input}$  ;
              Receive data  $x_{in}$  ,  $x'_{in}$  ,  $y_{in}$  ;
              Store data into the input port ;
              Activate FUL signal ;
            & Send  $REQ_{output}$  to a destined PE ;
              Wait for  $ACK_{output}$  from the destined PE ;
              Send  $x_{in}$  to output port & buffer-for-add ;
            & Add  $(v_{i,j} \leftarrow x_{in} + x'_{in})$  ;
              Store  $v_{i,j}$  into buffer-for-mult ;
              Read  $\omega_i$  from buffer-for-mult ;
                &  $v_{i,j}$  from buffer-for-mult ;
              Mult  $(u_{i,j} \leftarrow \omega_i v_{i,j})$  ;
              Store  $u_{i,j}$  into buffer-for-add ;
              Read  $u_{i,j}$  from buffer-for-add ;
                &  $y_{in}$  from buffer-for-add ;
              Add  $(y_{out} \leftarrow u_{i,j} + y_{in})$  ;
              Store  $y_{out}$  into buffer-for-W ;
            & Activate FLG signal ;
          End While
        End For
      End While
    End
  
```

- I FULL INPUT BUFFERS FLAG
- F INTERNAL FLAG
- A ACKNOWLEDGE FLAG
- E EMPTY-INPUT BUFFER FLAG
- ADD ADDITION TRANSITION
- MULT MULTIPLICATION TRANSITION

- TRANSITION (I/O) LATCH
- DATA LINE
- SIGNAL (ACK/REQ) LINE
- DATA BUFFER
- R REQUEST FLAG

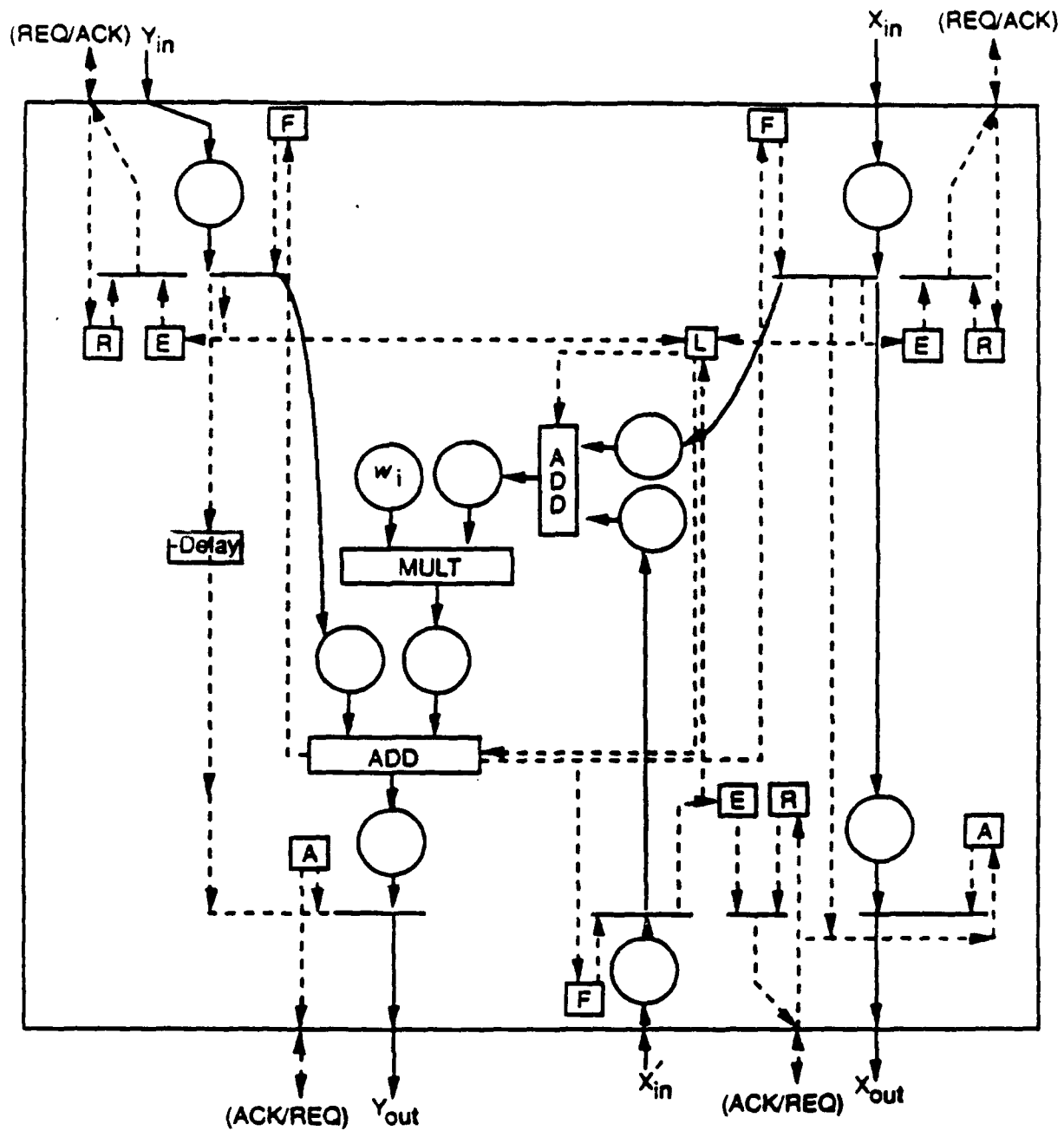


Fig. 10. Proposed protocol in (type I) $PE_{i,j}$.

2.3 Performance Aspect

The execution time modeling of an asynchronous system can represent quantitative evaluation of the developed algorithms. We developed three performance measures, *throughput*, *speed up*, and *efficiency*, for evaluating the asynchronous spiral SA realizations. The performance analysis technique is next demonstrated for the **LPF digital filter** asynchronous SA realization [16]. The following notations will be used to describe the time models of the arrays:

- T_{In} = time for reading data from input port
- T_{D_i} = time for internal data transfer
- T_{D_o} = time for external data transfer
- T_R = signal REQ propagation time
- T_{A_k} = signal ACK propagation time
- T_{Fire} = time for firing data via latch
- T_C = clock time for synchronous systolic array realization
- T_E = signal EMP propagation time
- T_{A_d} = computing time for addition
- T_M = computing time for multiplication
- T_W = time for fetching data from buffer-for-add/multiply/etc.
- T_F = signal FLG/FUL propagation time
- T_{Out} = time for reading data from output port

The time delays, starting with the reading of data from input port, for the respective asynchronous spiral SA and the corresponding synchronous systolic array can be described by the following relationships (see Figure 10):

$$T_{PE_{(asyn)}} = \max\{(T_{In} + T_{D_i} + (T_R + T_{Fire}) + (T_{A_k} + T_{Fire}) + T_{Out} + T_{D_o}), \\ ((T_{In} + T_{D_i} + T_W + T_{A_d} + T_W + T_M + T_W + T_{A_d} + T_W) + (T_R + T_{Fire}) + \\ (T_{A_k} + T_{Fire}) + T_{Out} + T_{D_o})\} \quad (13)$$

$$T_{total_{(asyn)}} = (U)(T_{In} + T_{D_i} + T_{Out} + T_{D_o} + 4T_W + 2T_{A_d} + T_M + T_{Com}) \quad (14)$$

where,

$$T_{Com} = T_{A_k} + 2T_{Fire} + T_R$$

and,

$$T_{total_{(syn)}} = M'(T_{In} + T_{D_i} + T_{D_o} + 3(\max\{T_{A_d}, T_M\} + T_W)) \\ = M'(T_{In} + T_{D_i} + T_{D_o} + 3(T_M + T_W)) = M'T_C \quad (15)$$

where, M' refers to the number of PEs in the synchronous systolic array, and

$$M' = \begin{cases} M + 1 & \text{if } N \text{ odd, even symmetry} \\ M & \text{otherwise} \end{cases}$$

The *throughput* can be obtained by dividing the number of processors by the execution time per sample. Thus, the *throughput* using $M \times U$ processors, $R(M_n)$ is defined by the following equation:

$$R(M_n) = \frac{M}{T_{In} + T_{D_i} + T_{Out} + T_{D_o} + 4T_W + 2T_{A_d} + T_M + T_{Com}} \quad (16)$$

The *speedup* is the ratio of the *throughput* in the asynchronous spiral SA system to that in a reference synchronous system. From above, the *speedup* via an asynchronous spiral SA realization is represented as

$$S(M_n) = \frac{M (T_{In} + T_{D_i} + T_{D_o} + 3 (T_M + T_W))}{M' (T_{In} + T_{D_i} + T_{Out} + T_{D_o} + 4T_W + 2T_{A_d} + T_M + T_{Com})} \quad (17)$$

The system *efficiency* is defined as the ratio of the *speedup* and the number of processors

$$E(M_n) = \frac{T_{In} + T_{D_i} + T_{D_o} + 3 (T_M + T_W)}{U(M') (T_{In} + T_{D_i} + T_{Out} + T_{D_o} + 4T_W + 2T_{A_d} + T_M + T_{Com})} \quad (18)$$

The above equation indicates that the reduction of interprocessor communication time, T_{Com} , greatly improves the system's *efficiency* [16].

2.4 References

- [1] Kung, H. T., and C. E. Leiserson, "Systolic Arrays (for VLSI)." In *Sparse Matrix Proc.*; Duff et al., Eds., 1979, pp. 256-282.
- [2] Kung, H. T., and C. E. Leiserson, "Algorithms for VLSI Processor Arrays." In *Introduction to VLSI Systems*; C. Mead and L. Conway, Eds. Reading, MA: Addison-Wesley, 1980, pp. 263-332.
- [3] Melkemi, L., and M. Tchuenté, "Algebraic and Combinatorial Aspects of Systolic Algorithms." In *Proc. of Int. Workshop on Parallel Algorithms and Arch.*, 1986, pp. 269-279.
- [4] Cytron, R. G., *Compile-Time Scheduling and Optimization for Asynchronous Machines*. Ph.D. Dissertation, University of Illinois at Urbana-Champaign, 1985.
- [5] Peng, S., and M. Jun, "A New VLSI 2-D Systolic Array for Matrix Multiplication and Its Applications." In *Proc. of Int. Conf. on Parallel Processing*, 1988, pp. 169-172.
- [6] Kung, H. T. "Why Systolic Architectures?" *Computer*, Jan. 1982, Vol. 15, No. 1, pp. 37-46.

- [7] Kung, S. Y., and S. C. Lo, "Supercomputing with Systolic/Wavefront Array Processors." In *IEEE Proc.*, July 1984, Vol. 72, No. 7.
- [8] Mills, P. L., "The Design of Bit Parallel Systolic Filter Algorithms." In *Proc. ACM Computer Science Conference*, 1984, pp. 121-130.
- [9] Cooley, J. W., et al., "The Fast Fourier Transformation Algorithm: Programming Considerations in the Calculation of Sine, Cosine, and Laplace Transformations." *Digital Signal Processing*; L. R. Rabiner and C. M. Rader, Eds. New York, NY: IEEE Press, 1972, pp. 271-293.
- [10] Oppenheim, A. V., and R. W. Schaffer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.
- [11] Kung, S. Y., *VLSI Array Processors*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
- [12] Kaminski, P. G., A. E. Bryson, and S. F. Schmidt, "Discrete Square-Root Filtering: A Survey of Current Techniques." In *IEEE Trans. on Automatic Control*, Dec. 1971, Vol. AC-16, No. 6, pp. 727-735.
- [13] Griffiths, L. J., "A Continuously-Adaptive Filter Implemented as a Lattice Structure." In *Proc. ICASSP-77*, May 1977, pp. 683-686.
- [14] Haykin, S., *Adaptive Filter Theory*. Second Edition, Prentice-Hall, 1991
- [15] Widrow, B., and S. Sterns, *Adaptive Signal Processing*. Prentice-Hall, 1985.
- [16] Reihani, K., and Y. Fan, "Spiral Systolic Design with Asynchronous Controls for LPF Digital Filters." Under review by the *IEEE Transactions on Signal Processing*.
- [17] Reihani, K., and M. Mayorga, "Asynchronous Systolic Realization for One-Step Adaptive Lattice Predictors." To appear in the *Proc. of the Fourth Int. Conf. on Signal Processing Applications and Technology - ICSPAT '93*, Santa Clara, CA, Sept. 1993.
- [18] Reihani, K., and M. Mayorga, "Asynchronous Systolic Realization for Arbitrarily Large LMS Adaptive Filters." Under Review by the *IEEE Transactions on Computers*.
- [19] Reihani, K., and Z. Zhang, "Asynchronous and Concurrent Systolic Array Design for Discrete Hadamard Transforms." PSL Internal Report, June 1992.
- [20] Reihani, K., and Y. Fan, "Spiral Systolic Design with Asynchronous Controls for DFT and IDFT." PSL Internal Report, May 1993.

3. PUBLICATIONS AND REPORTS

- [1] Reihani, K., and Y. Fan, "Spiral Systolic Design with Asynchronous Controls for LPF Digital Filters." Under review by the *IEEE Transactions on Signal Processing*.
- [2] Reihani, K., and M. Mayorga, "Asynchronous Systolic Realization for One-Step Adaptive Lattice Predictors." To appear in the *Proc. of the Fourth Int. Conf. on Signal Processing Applications and Technology - ICSPAT '93*, Santa Clara, CA, Sept. 1993.
- [3] Reihani, K., and M. Mayorga, "Asynchronous Systolic Realization for Arbitrarily Large LMS Adaptive Filters." Under Review by the *IEEE Transactions on Computers*.
- [4] Reihani, K., and Z. Zhang, "Asynchronous and Concurrent Systolic Array Design for Discrete Hadamard Transforms." PSL Internal Report, June 1992.
- [5] Reihani, K., and Y. Fan, "Spiral Systolic Design with Asynchronous Controls for DFT and IDFT." PSL Internal Report, May 1993.

4. PERSONNEL SUPPORTED AND DEGREES AWARDED

Major Investigator

Reihani, K.

Research Area

Signal Processing and Image Understanding

Doctoral Students

Fan, Yiping

Dissertation

Current Ph.D. Student: Signal Processing

Zhang, Zhi

Current Ph.D. Student: Signal Processing

APPENDIX A

THEORETICAL ASPECT FOR DSP ALGORITHMS

— Arbitrarily Large LMS Adaptive Filters

A standard form for the LMS algorithm when real discrete time signals, x_j , are used is described by

$$W_{j+1} = W_j + 2\mu \xi_j X_j \quad (A1)$$

where W_j and X_j are vectors representing, respectively, the filter weights and the outputs of the transversal filter line at time j

$$W_j^T = [w_{1,j}, w_{2,j}, \dots, w_{n,j}]$$

$$X_j^T = [x_j, x_{j-1}, \dots, x_{j-n+1}]$$

and μ is the gain constant that regulates the speed and stability of adaptation. The error, ξ_j , is the difference between the desired response d_j , an externally supplied training signal, and the filter output y_j

$$\xi_j = d_j - y_j$$

where the filter output is given by

$$y_j = X_j^T W_j$$

— Convolution and Correlation

Convolution and correlation are closely related operations that are basic to many areas of digital signal processing. The convolution of two time series is equivalent to the product of the Fourier transforms of the two time series. They can be expressed for a finite length N as

$$\text{Convolution: } c_{xy}(n) = \text{conv}\{[x(k), y(k)]\} = \sum_{k=0}^{N-1} x(k)y(n-k) = \sum_{k=0}^{N-1} x(n-k)y(k) \quad (A2)$$

$$\text{Correlation: } r_{xy}(n) = \text{corr}\{[x(k), y(k)]\} = \sum_{k=0}^{N-1} x(k)y(n+k) = \sum_{k=0}^{N-1} x(n+k)y(k) \quad (A3)$$

— DFT and IDFT

The discrete Fourier transform (DFT) is used to transform an ordered sequence of data samples, usually from the time domain into the frequency domain, so that spectral information about the sequence can become known explicitly. The inverse discrete Fourier Transform (IDFT) is used to obtain a data sequence in time domain

from its complex spectrum. If we denote $x(k)$ ($k = 0, 1, \dots, N-1$) as a time series and $X(k)$ as the output of an N point Fourier transform, then

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j \frac{2\pi n}{N} k} = \sum_{n=0}^{N-1} x(n) w(nk) \quad k = 0, 1, \dots, N-1 \quad (A4)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j \frac{2\pi n}{N} k} = \frac{1}{N} \sum_{k=0}^{N-1} X(k) w(-nk) \quad n = 0, 1, \dots, N-1 \quad (A5)$$

where

$$w(nk) = e^{-j \frac{2\pi n}{N} k}$$

$$w(-nk) = e^{j \frac{2\pi n}{N} k}$$

— DHT

A Hadamard matrix is a matrix whose columns are orthogonal and comprised of elements whose value is either -1 or +1. Given a Hadamard matrix of order N , Hadamard matrices of higher order may be generated by the simple recursive formulation.

$$H_{2N} = \left[\begin{array}{c|c} H_N & H_N \\ \hline -H_N & -H_N \end{array} \right] \quad (A6)$$

The Hadamard matrix of the lowest order becomes

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (A7)$$

and the Hadamard matrix of order $N = 4$ is given as

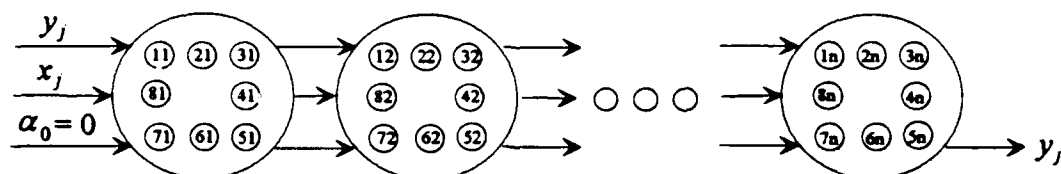
$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} \quad (A8)$$

APPENDIX B

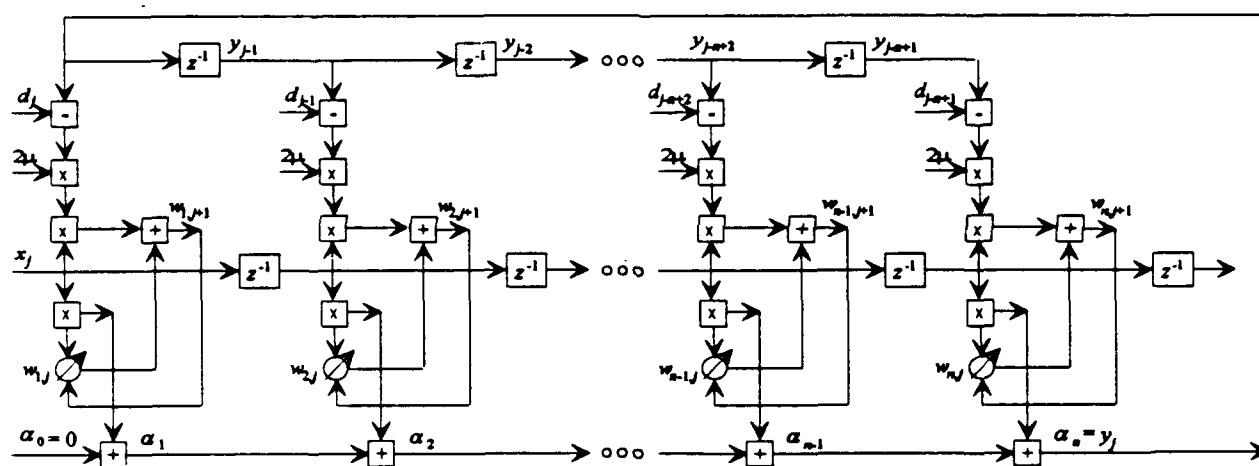
SYSTEM ASPECT I(A)

— Arbitrarily Large LMS Adaptive Filters

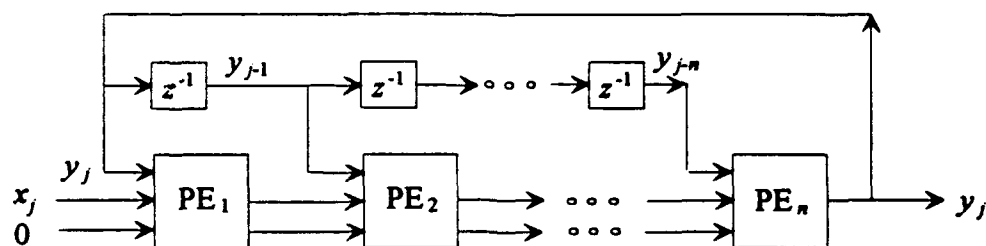
— Computational graph for y_j



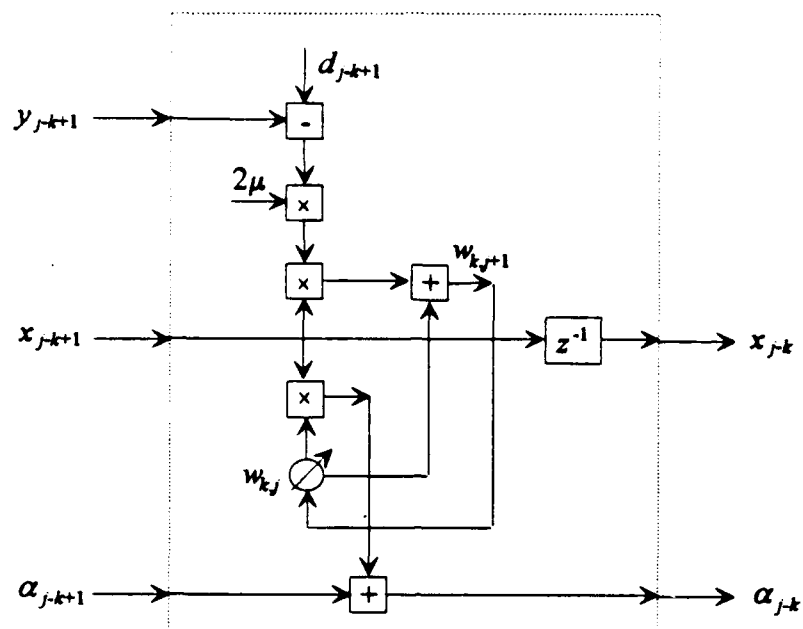
— Computational scheme for y_j



— Interconnection scheme for y_j

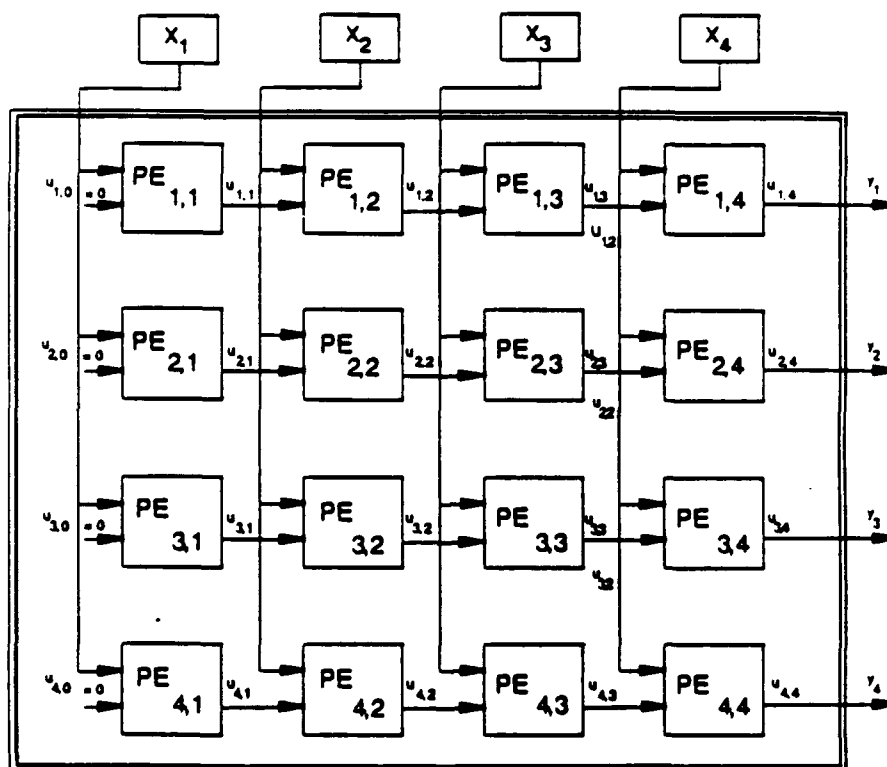


— Proposed PE_k for y_j



— DHT

— Systolic array scheme for a 4 x 4 DHT

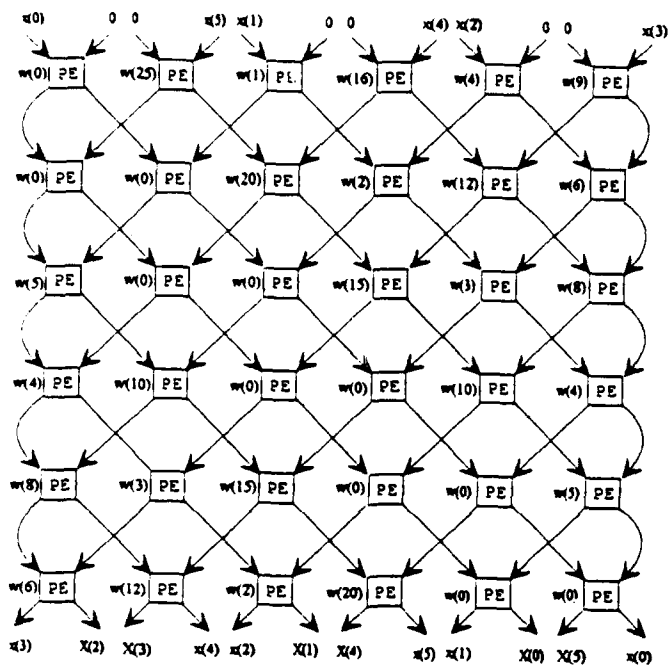


APPENDIX C

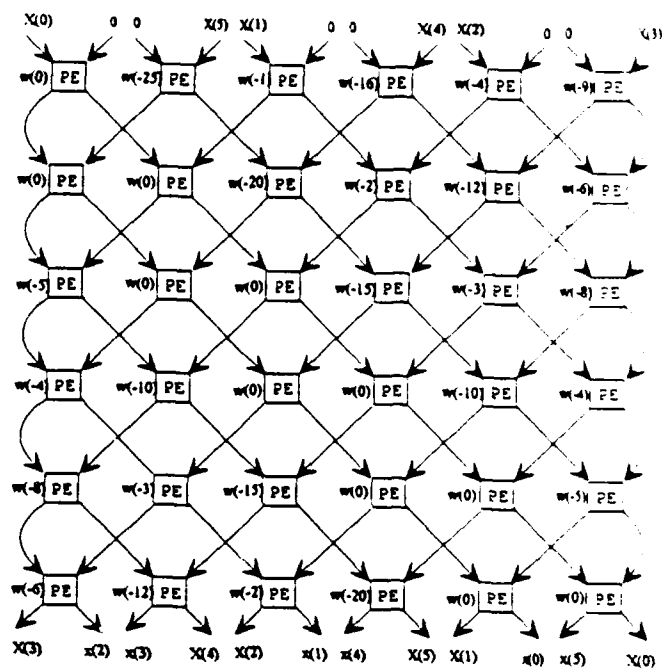
SYSTEM ASPECT I(B)

— Spiral SA Structure to Compute Equations A4 and A5 (N=6)

— DFT and IDFT



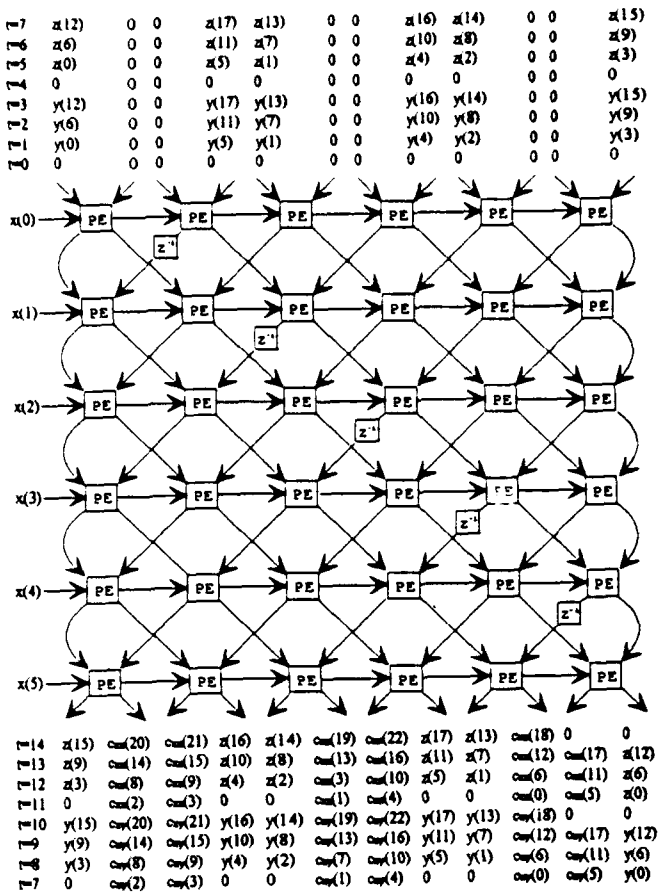
DFT



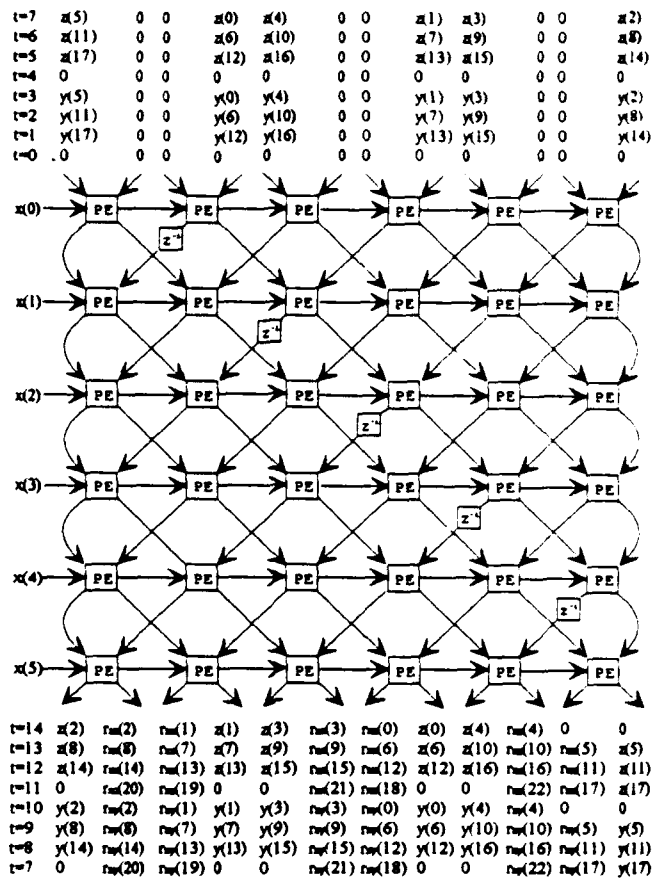
IDFT

— Spiral SA Structure to Compute Equations A2 and A3 ($N = 6$)

— Convolution and correlation



Convolution

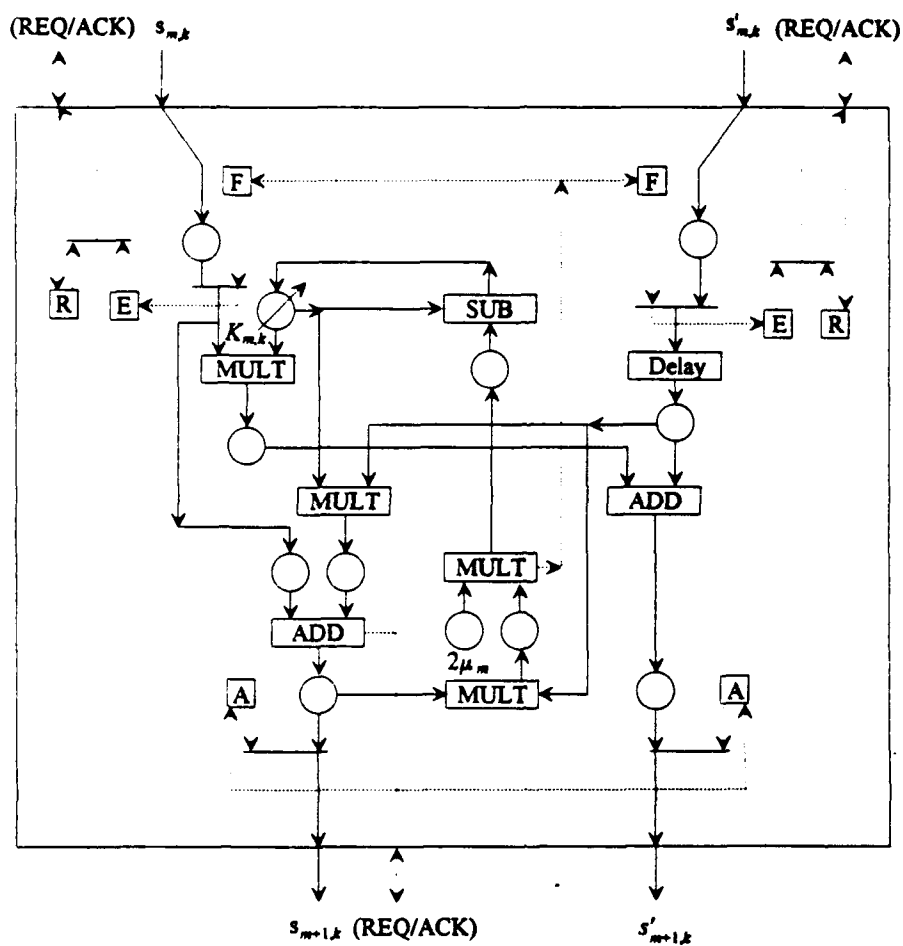
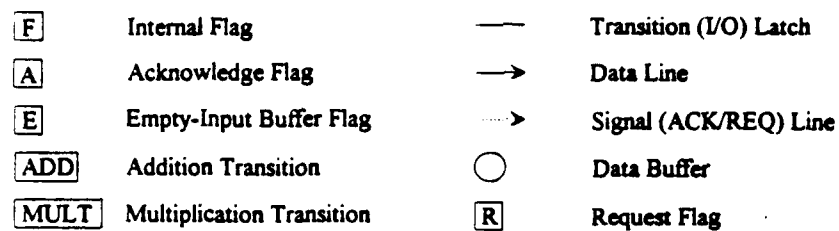


Correlation

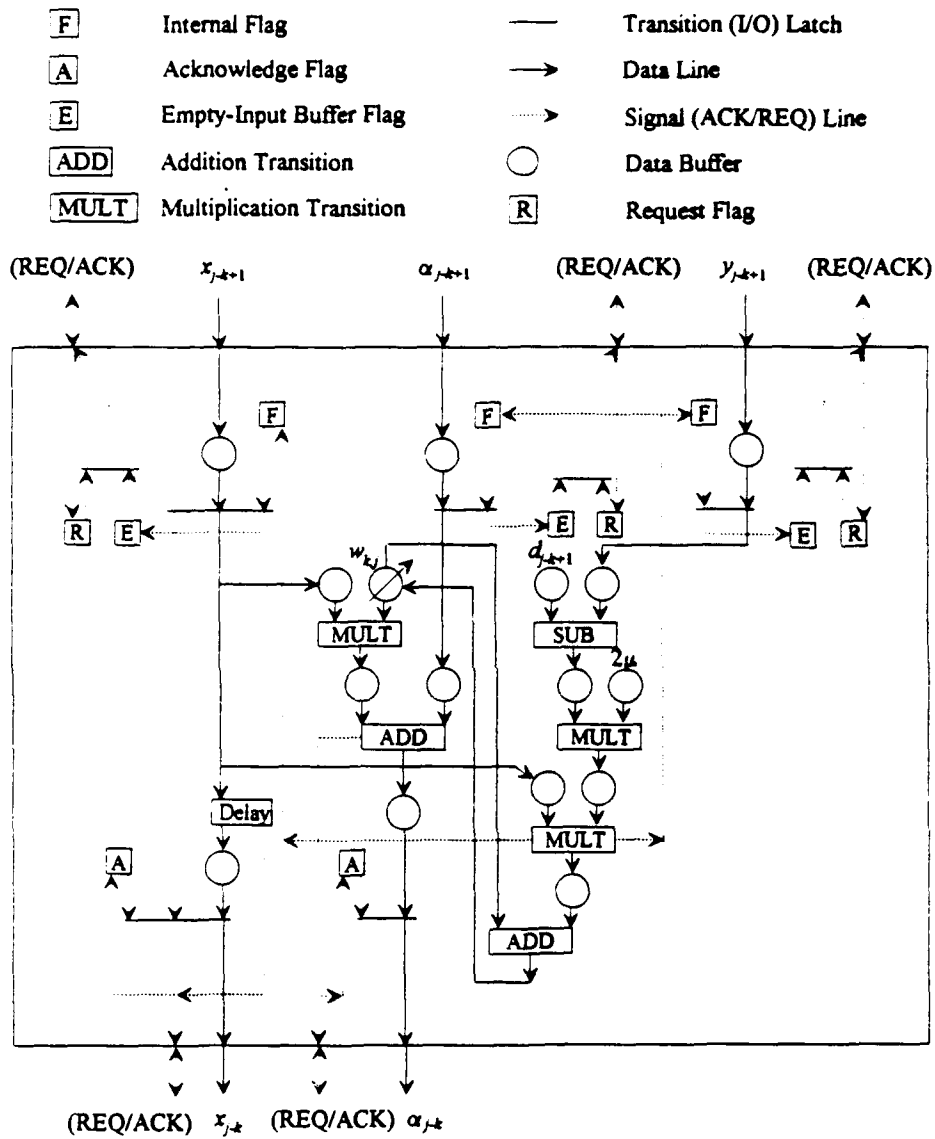
APPENDIX D

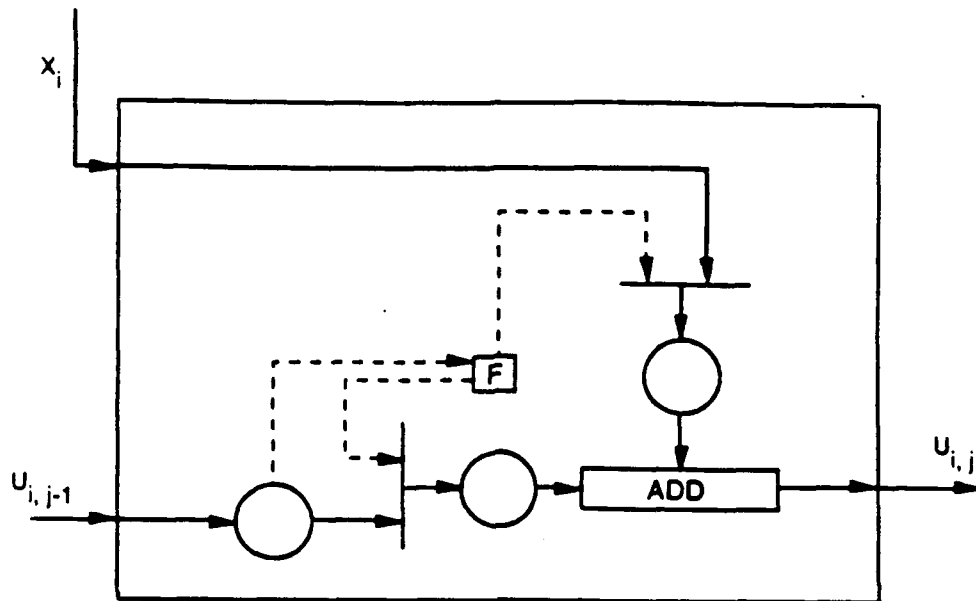
SYSTEM ASPECT II

— One-Step Adaptive Lattice Predictors' PE_m Protocol

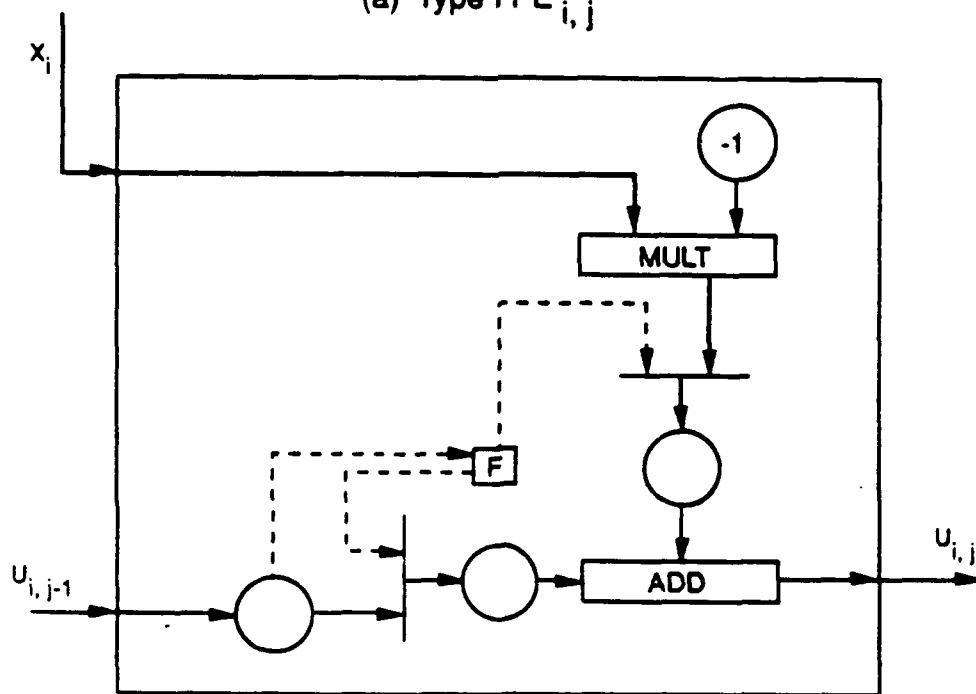


— Arbitrarily Large LMS Adaptive Filters' PE_k Protocol





(a) Type I PE_{i,j}



(b) Type II PE_{i,j}